

A MULTIVARIATE FAST DISCRETE WALSH TRANSFORM WITH AN APPLICATION TO FUNCTION INTERPOLATION

KWONG-IP LIU, JOSEF DICK, AND FRED J. HICKERNELL

ABSTRACT. For high dimensional problems, such as approximation and integration, one cannot afford to sample on a grid because of the curse of dimensionality. An attractive alternative is to sample on a low discrepancy set, such as an integration lattice or a digital net. This article introduces a multivariate fast discrete Walsh transform for data sampled on a digital net that requires only $\mathcal{O}(N \log N)$ operations, where N is the number of data points. This algorithm and its inverse are digital analogs of multivariate fast Fourier transforms.

This fast discrete Walsh transform and its inverse may be used to approximate the Walsh coefficients of a function and then construct a spline interpolant of the function. This interpolant may then be used to estimate the function's effective dimension, an important concept in the theory of numerical multivariate integration. Numerical results for various functions are presented.

1. INTRODUCTION

The idea of the fast Fourier transforms goes back to Gauss and has been popular ever since the seminal work of Cooley and Tukey [5]. Let f be a function from $[0, 1]$ to the complex numbers. The task is to compute

$$\tilde{f}(k) = \sum_{n=0}^{N-1} f(n/N) e^{2\pi i k n / N} \quad \text{for } k = 0, \dots, N-1.$$

These are N sums, each consisting of N summands. Hence a straight forward calculation would have complexity of $\mathcal{O}(N^2)$ operations. But the sums have a certain structure which can be exploited. Indeed Cooley and Tukey showed that those sums can be computed with $\mathcal{O}(N \log N)$ operations. (There is some dependence of the implied constant on the number N ; the algorithm works best if N is a prime power, see [5]).

In higher dimensions an effect commonly referred to as *the curse of dimensionality* occurs. Let $f : [0, 1]^s \rightarrow \mathbb{C}$ and consider the discrete Fourier transform

$$\tilde{f}(\mathbf{k}) = \sum_{n_1, \dots, n_s=0}^{p-1} f(n_1/p, \dots, n_s/p) e^{2\pi i (n_1 k_1 + \dots + n_s k_s) / p}$$

Received by the editor August 4, 2008.

2000 *Mathematics Subject Classification.* 42C10, 41A15.

This research was supported in part by the Hong Kong Research Grants Council grant HKBU/2009/04P, the HKSAR Research Grants Council Project No. HKBU200605 and the United States National Science Foundation grant NSF-DMS-0713848.

for all $\mathbf{k} = (k_1, \dots, k_s) \in \{0, \dots, p-1\}^s$. Here the number of points sampled is $N = p^s$. Hence if s is large, say 100 for example, then choosing even $p = 2$ yields $N = 2^{100} \approx 10^{30}$ points, making such a computation infeasible for contemporary computers.

In the example above the design or set of sample points is a grid aligned with the coordinate axes, $\{(n_1/p, \dots, n_s/p) : 0 \leq n_j < p\}$. To avoid the curse of dimensionality one needs a much smaller point set that is constructed differently. Such point sets have previously been considered in the context of numerical integration, see [14, 19]. Two popular construction methods are integration lattices (see [19]) and digital nets and sequences (see [14]). This article focuses on the latter family of points. (Numerical approximation using lattice rule designs and an FFT has been treated in [12, 29].) The first examples of digital sequences were given by Sobol [20] and Faure [9] before Niederreiter introduced the general concept of (t, m, s) -nets and (t, s) -sequences. See [15] for a recent survey. These constructions yield extremely well distributed point sets if the quality parameter t is small. Digital (t, m, s) -nets are a special construction of (t, m, s) -nets, and in the same way, digital (t, s) -sequences are a special construction of (t, s) -sequences. Digital constructions are introduced below.

Definition 1.1. Let \mathbb{Z}_p be a finite field of prime order p , let C_1, \dots, C_s be s $m \times m$ -matrices over $\mathbb{Z}_p = \{0, 1, \dots, p-1\}$. The digital (t, m, s) -net $P(\mathcal{C}) = \{\mathbf{x}_0, \dots, \mathbf{x}_{p^m-1}\}$, based on $\mathcal{C} = (C_1, \dots, C_s)$, is then defined as follows: let $0 \leq n < p^m$ and $n = n_0 + n_1p + \dots + n_{m-1}p^{m-1}$ be the base p representation of n . Define $\vec{n} = (n_0, \dots, n_{m-1})^T \in \mathbb{Z}_p^m$ and let

$$(1.1) \quad \vec{y}_{j,n} = C_j \vec{n} \in \mathbb{Z}_p^m.$$

Express $\vec{y}_{j,n}$ as $(y_{j,n,1}, \dots, y_{j,n,m})^T \in \mathbb{Z}_p^m$, and then define

$$x_{j,n} = y_{j,n,1}p^{-1} + \dots + y_{j,n,m}p^{-m}.$$

The n -th point \mathbf{x}_n of the digital net $P(\mathcal{C})$ over the finite field \mathbb{Z}_p is given by $\mathbf{x}_n = (x_{1,n}, \dots, x_{s,n})$.

The t value is a non-negative integer such that for all $0 \leq d_1, \dots, d_s \leq m-t$ with $d_1 + \dots + d_s = m-t$ the system of vectors $c_1^{(1)}, \dots, c_{d_1}^{(1)}, \dots, c_1^{(s)}, \dots, c_{d_s}^{(s)}$ is linearly independent over \mathbb{Z}_p . Here $c_k^{(j)}$ refers to the k -th row of the matrix C_j . For a geometrical interpretation of the t value see, for example, [14]. Smaller values of t characterize more uniformly distributed nets.

Digital nets are often used in conjunction with certain wavelets, namely Haar functions, first used by Sobol [21], and Walsh functions, first used by Larcher [10] and Larcher and Traunfellner [11].

Next, Walsh functions, which are piecewise constant, in base p are defined. For more information on Walsh functions see for example [4, 24] (or in the context of numerical integration see [6]). Throughout this article let \mathbb{N}_0 denote the set of non-negative integers and let \mathbb{N} denote the set of positive integers.

Definition 1.2. Let $p \geq 2$ be an integer. For a non-negative integer wavenumber k with base p representation

$$k = k_0 + k_1p + \dots + k_{a-1}p^{a-1},$$

with $k_i \in \mathbb{Z}_p$, the Walsh function ${}_p\text{wal}_k : [0, 1) \longrightarrow \mathbb{C}$ is defined by

$${}_p\text{wal}_k(x) := \omega_p^{x_1 k_0 + \dots + x_a k_{a-1}},$$

where $\omega_p = e^{2\pi i/p}$, for $x \in [0, 1)$ with base p representation $x = x_1/p + x_2/p^2 + \dots$ (unique in the sense that infinitely many of the x_i must be different from $p-1$). If it is clear which base p is chosen we simply write wal_k .

Definition 1.3. For dimension $s \geq 2$, $x_1, \dots, x_s \in [0, 1)$ and $k_1, \dots, k_s \in \mathbb{N}_0$ define ${}_p\text{wal}_{k_1, \dots, k_s} : [0, 1)^s \longrightarrow \mathbb{C}$ by

$${}_p\text{wal}_{k_1, \dots, k_s}(x_1, \dots, x_s) := \prod_{j=1}^s {}_p\text{wal}_{k_j}(x_j).$$

For wavenumber vectors $\mathbf{k} = (k_1, \dots, k_s) \in \mathbb{N}_0^s$ and $\mathbf{x} = (x_1, \dots, x_s) \in [0, 1)^s$ we write

$${}_p\text{wal}_{\mathbf{k}}(\mathbf{x}) := {}_p\text{wal}_{k_1, \dots, k_s}(x_1, \dots, x_s).$$

Again, if it is clear which base is meant we simply write $\text{wal}_{\mathbf{k}}(\mathbf{x})$.

Let a Walsh series $f \in \mathcal{L}_2([0, 1]^s)$ be defined by

$$f(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{N}_0^s} \hat{f}(\mathbf{k}) \text{wal}_{\mathbf{k}}(\mathbf{x}),$$

where the Walsh coefficients are given by

$$(1.2) \quad \hat{f}(\mathbf{k}) = \int_{[0, 1]^s} f(\mathbf{x}) \overline{\text{wal}_{\mathbf{k}}(\mathbf{x})} d\mathbf{x},$$

since the Walsh functions are mutually orthonormal. Here \bar{a} denotes the complex conjugate of a complex number a .

The aim now is to approximate Walsh coefficients of a function f with wavenumbers lying in a certain set of wavenumbers, $\mathcal{K}(\mathcal{C})$, depending on the digital net $P(\mathcal{C})$ defined by the generating matrices $\mathcal{C} = (C_1, \dots, C_s)$. The details of how $\mathcal{K}(\mathcal{C})$ is chosen is explained in the next section. A digital net with p^m points can be used to estimate $|\mathcal{K}(\mathcal{C})| = p^m$ Walsh coefficients, where $|\cdot|$ denotes the cardinality of a set. For $\mathbf{k} \in \mathcal{K}(\mathcal{C})$ we approximate $\hat{f}(\mathbf{k})$ by the finite sum

$$(1.3) \quad \tilde{f}(\mathbf{k}) = \frac{1}{p^m} \sum_{\mathbf{x} \in P(\mathcal{C})} f(\mathbf{x}) \overline{\text{wal}_{\mathbf{k}}(\mathbf{x})}.$$

We call $\tilde{f}(\mathbf{k})$ the discrete Walsh coefficients because (1.3) is just a discrete version of (1.2). Those discrete Walsh coefficients provide us with valuable information about the function at hand.

A naive calculation of the p^m discrete Walsh coefficients $\tilde{f}(\mathbf{k})$ with $\mathbf{k} \in \mathcal{K}(\mathcal{C})$ would require $\mathcal{O}(p^{2m})$ operations, but using the fast discrete Walsh transform algorithm described in the following section, we can reduce it to $\mathcal{O}(mp^m \log p)$ operations. In Section 3 the discrete Walsh coefficients are used to interpolate functions based on observations on the digital net design. The inverse discrete Walsh transform then provides an interpolatory approximation of the original function. The ANOVA decomposition of this interpolation provides information about the effective dimension of the function as explained in Section 4. In the last section the fast discrete Walsh transform is used to approximate Walsh coefficients and effective dimensions of some explicit test functions.

2. MULTIVARIATE FAST DISCRETE WALSH TRANSFORM OVER DIGITAL NETS

In this section we introduce an FFT-like algorithm for a multivariate fast discrete Walsh transform over digital nets. An essential role is played by the dual net of a digital net.

2.1. The Dual Net. Let $\mathcal{C} = (C_1, \dots, C_s)$ be the vector of matrices generating a digital (t, m, s) -net $P(\mathcal{C})$ over a finite field \mathbb{Z}_p , where p is prime. For any wavenumber vector $\mathbf{k} = (k_1, \dots, k_s) \in \mathbb{N}_0^s$, we define

$$\mathcal{C} \cdot \mathbf{k} = C_1^T \vec{k}_1^{(m)} + \dots + C_s^T \vec{k}_s^{(m)} \in \mathbb{Z}_p^m,$$

where for $k_j = k_{j,0} + k_{j,1}p + \dots$ we define $\vec{k}_j^{(m)} = (k_{j,0}, \dots, k_{j,m-1})^T \in \mathbb{Z}_p^m$ as the m -element truncation of \vec{k}_j , and all operations are carried out in the finite field \mathbb{Z}_p . The dual net $\mathcal{D}(\mathcal{C})$ of the digital net $P(\mathcal{C})$ is the set of all wavenumbers which make this dot product zero:

$$\mathcal{D}(\mathcal{C}) = \{\mathbf{k} \in \mathbb{N}_0^s : \mathcal{C} \cdot \mathbf{k} = \vec{0}\}.$$

The dual net appears in the worst-case error for multivariate integration in certain Walsh spaces, see [6]. Therein the worst-case error is just the sum of a certain function over all elements in the dual net except $\mathbf{0}$.

The dual net satisfies

$$(2.1) \quad \frac{1}{p^m} \sum_{\mathbf{x} \in P(\mathcal{C})} \text{wal}_{\mathbf{k}}(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{k} \in \mathcal{D}(\mathcal{C}) \\ 0 & \text{otherwise,} \end{cases}$$

which was shown, for example, in [6]. Indeed, this property could also be used to define the dual net.

For $k \in \mathbb{N}_0$ with $k = k_0 + k_1p + \dots$ let $\nu(0) = 0$ and for $k > 0$ let $\nu(k) = 1 + \max\{i : k_i \neq 0\}$. For $\mathbf{k} = (k_1, \dots, k_s) \in \mathbb{N}_0^s$ define $\nu(\mathbf{k}) = \sum_{i=1}^s \nu(k_i)$ (see [13, 16]). The function ν is a norm on the elements in the wavenumber space. It depends on the most significant bit of the coordinates and can be related to the t -value of a digital net, see [16].

For non-negative integers $k, l \in \mathbb{N}_0$ with $k = k_0 + k_1p + \dots$ and $l = l_0 + l_1p + \dots$ a digit-wise addition and subtraction in base p can be defined by $k \oplus l = a_0 + a_1p + \dots$ where $a_i \equiv k_i + l_i \pmod{p}$ and $k \ominus l = b_0 + b_1p + \dots$ where $b_i \equiv k_i - l_i \pmod{p}$. For non-negative integer vectors the digit-wise addition and subtraction are defined component-wise.

Using this digit-wise addition and subtraction we obtain a group structure on \mathbb{N}_0^s of which the dual net $\mathcal{D}(\mathcal{C})$ forms a subgroup. It can be checked that the cosets of the subgroup $\mathcal{D}(\mathcal{C})$ are given by

$$\mathcal{D}(\vec{h}) = \{\mathbf{k} \in \mathbb{N}_0^s : \mathcal{C} \cdot \mathbf{k} = \vec{h}\}$$

for $\vec{h} \in \mathbb{Z}_p^m$ and hence there are p^m cosets. Note that $\mathcal{D}(\vec{0}) = \mathcal{D}(\mathcal{C})$, i.e., the coset containing $\vec{0}$ is the dual net. The set of wavenumbers whose Walsh coefficients are to be approximated, $\mathcal{K}(\mathcal{C})$, is then obtained by choosing exactly one representative in each coset. For each $\vec{h} \in \mathbb{Z}_p^m$ identify $\mathbf{k} \in \mathcal{D}(\vec{h})$ such that $\nu(\mathbf{k}) \leq \nu(\mathbf{l})$ for all $\mathbf{l} \in \mathcal{D}(\vec{h})$. This \mathbf{k} is the representative of $\mathcal{D}(\vec{h})$ chosen to be in $\mathcal{K}(\mathcal{C})$. In the case of more than one \mathbf{k} from the same coset satisfying this condition, one may choose, for

example, the \mathbf{k} that is the smallest in lexicographic order. That is,

$$\mathcal{K}(\mathcal{C}) = \left\{ \mathbf{k} \in \mathbb{N}_0^s : \mathbf{k} \in \mathcal{D}(\vec{h}) \text{ for some } \vec{h} \in \mathbb{Z}_p^m, \text{ and for any } \mathbf{l} \in \mathcal{D}(\vec{h}) \right.$$

we have $\nu(\mathbf{k}) \leq \nu(\mathbf{l})$ and if $\nu(\mathbf{k}) = \nu(\mathbf{l})$ for some $\mathbf{l} \in \mathcal{D}(\vec{h})$, then

$$k_1 = l_1, \dots, k_{j-1} = l_{j-1}, k_j < l_j \text{ for some } j = 1, \dots, s. \}$$

This definition implies that the zero vector $\mathbf{0}$ is automatically in $\mathcal{K}(\mathcal{C})$, and that \mathbb{N}_0^s is the direct sum of $\mathcal{K}(\mathcal{C})$ and $\mathcal{D}(\mathcal{C})$.

2.2. Multivariate fast discrete Walsh transform over digital nets. Let a Walsh series $f \in \mathcal{L}_2([0, 1]^s)$ be given by

$$f(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{N}_0^s} \hat{f}(\mathbf{k}) \text{wal}_{\mathbf{k}}(\mathbf{x}).$$

For all $\mathbf{k} \in \mathcal{K}(\mathcal{C})$ one may approximate $\hat{f}(\mathbf{k})$ by the discrete Walsh transform (DWT), $\tilde{f}(\mathbf{k})$, which is defined as

$$\tilde{f}(\mathbf{k}) = \frac{1}{p^m} \sum_{\mathbf{x} \in P(\mathcal{C})} f(\mathbf{x}) \overline{\text{wal}_{\mathbf{k}}(\mathbf{x})} = \sum_{\mathbf{l} \in \mathbb{N}_0^s} \hat{f}(\mathbf{l}) \frac{1}{p^m} \sum_{\mathbf{x} \in P(\mathcal{C})} \text{wal}_{\mathbf{l}}(\mathbf{x}) \overline{\text{wal}_{\mathbf{k}}(\mathbf{x})}.$$

Note that $\text{wal}_{\mathbf{l}}(\mathbf{x}) \overline{\text{wal}_{\mathbf{k}}(\mathbf{x})} = \text{wal}_{\mathbf{l} \ominus \mathbf{k}}(\mathbf{x})$ (see [6]), and hence the rightmost sum in the equation above is one if $\mathbf{l} \ominus \mathbf{k} \in \mathcal{D}(\mathcal{C})$ and zero otherwise. Thus it follows that

$$(2.2) \quad \tilde{f}(\mathbf{k}) = \hat{f}(\mathbf{k}) + \sum_{\mathbf{l} \in \mathcal{D}(\mathcal{C}) \setminus \{\mathbf{0}\}} \hat{f}(\mathbf{k} \oplus \mathbf{l}).$$

Hence, the terms $\hat{f}(\mathbf{k} \oplus \mathbf{l})$ are completely aliased with each other for all $\mathbf{l} \in \mathcal{D}(\mathcal{C})$. We have chosen $\mathbf{k} \in \mathcal{K}(\mathcal{C})$ such that \mathbf{k} is closest to $\mathbf{0}$. Hence if higher frequency contributions are sufficiently small, that is, $\hat{f}(\mathbf{k})$ decays sufficiently fast the further \mathbf{k} is away from $\mathbf{0}$ with respect to the norm ν , then $\tilde{f}(\mathbf{k}) \approx \hat{f}(\mathbf{k})$.

A straightforward calculation of the discrete Walsh coefficients would require $\mathcal{O}(p^{2m})$ operations, as we have p^m sums to compute (one sum for each $\mathbf{k} \in \mathcal{K}(\mathcal{C})$) and each sum requires $\mathcal{O}(p^m)$ operations. But as shown below, certain parts in the summation above can be reused and thereby reducing the number of operations.

Let \mathbf{x}_n be the n -th point of the digital net $P(\mathcal{C})$ and let \mathbf{k} be the unique element of $\mathcal{D}(\vec{h}) \cap \mathcal{K}(\mathcal{C})$. Then we have

$$\text{wal}_{\mathbf{k}}(\mathbf{x}_n) = \omega_p^{\vec{n} \cdot \vec{h}},$$

because

$$\sum_{j=1}^s \sum_{i=1}^{\infty} y_{j,n,i} k_{j,i} = (\vec{k}_1^T C_1 + \dots + \vec{k}_s^T C_s) \vec{n} = \vec{h}^T \vec{n}.$$

Hence

$$\tilde{f}(\mathbf{k}) = \frac{1}{p^m} \sum_{n=0}^{p^m-1} f(\mathbf{x}_n) \omega_p^{-\vec{n} \cdot \vec{h}},$$

where $\vec{h} = \mathcal{C} \cdot \mathbf{k}$. The above sum may be written as

$$\tilde{f}(\mathbf{k}) = \frac{1}{p^m} \sum_{n_{m-1}=0}^{p-1} \omega_p^{-n_{m-1} h_{m-1}} \dots \sum_{n_1=0}^{p-1} \omega_p^{-n_1 h_1} \sum_{n_0=0}^{p-1} f(\mathbf{x}_n) \omega_p^{-n_0 h_0}.$$

Now, computing first the innermost sum for each $h_0 = 0, \dots, p-1$, then the second innermost sum for each $h_1 = 0, \dots, p-1$ and so on yields an algorithm which needs only $\mathcal{O}(mp^{m+1})$ operations. The details are given as follows.

Algorithm 1 (Fast Discrete Walsh transform (FWT)). For $n_0, \dots, n_{m-1} \in \mathbb{Z}_p$ we define $G^{(0)}(n_0, \dots, n_{m-1}) = f(\mathbf{x}_{n_0 + \dots + n_{m-1}p^{m-1}})$. Then for $r = 1, 2, \dots, m$ compute for all $n_r, \dots, n_{m-1} \in \mathbb{Z}_p$ and all $h_0, \dots, h_{r-1} \in \mathbb{Z}_p$ the sums

$$(2.3) \quad G^{(r)}(h_0, \dots, h_{r-1}, n_r, \dots, n_{m-1}) \\ = \sum_{n_{r-1}=0}^{p-1} \omega_p^{-n_{r-1}h_{r-1}} G^{(r-1)}(h_0, \dots, h_{r-2}, n_{r-1}, n_r, \dots, n_{m-1}).$$

For $\mathbf{k} \in \mathcal{K}(\mathcal{C})$ with $\mathcal{C} \cdot \mathbf{k} = \vec{h}$ let

$$\tilde{f}(\mathbf{k}) = \frac{1}{p^m} G^{(m)}(h_0, \dots, h_{m-1}).$$

Note that in each step one needs $\mathcal{O}(p^{m+1})$ operations, and as there are m steps, one needs $\mathcal{O}(mp^{m+1})$ operations altogether. Note also that the number of terms in the summation in the right side of (2.3) is p , which is a prime number. The index $n_{r-1} = 1, \dots, p-1$ (and h_{r-1}) forms a group under the multiplication modulo p . Thus, following the ideas of Rader's algorithm [18], we can rewrite the indices as $n_{r-1} = g^\alpha \pmod{p}$ and $h_{r-1} = g^\beta \pmod{p}$, where g is a primitive root of this group, and $\alpha = 0, \dots, p-2$, $\beta = 0, \dots, p-2$. By applying Rader's algorithm, we can further reduce the total number of steps to $\mathcal{O}(mp^m \log p)$ [18].

3. FUNCTION INTERPOLATION

In this section we consider multivariate spline interpolation over digital nets using the discrete Walsh coefficients described in the previous section. Multivariate spline interpolation over lattice rules was considered in [29]. See also [23] for more information on properties of splines.

3.1. Reproducing kernel Walsh space. Before we introduce the interpolation algorithm we introduce reproducing kernel Hilbert spaces based on Walsh functions, see [6]. In the following we define the weighted Hilbert space \mathcal{H}_K based on Walsh functions.

Consider the set of functions

$$\mathcal{H}_{0,K} = \left\{ f : f(\mathbf{x}) = \sum_{i=0}^{n'-1} \alpha_i K(\mathbf{x}, \mathbf{x}'_i) : n' \in \mathbb{N}_0, \alpha_i \in \mathbb{R}, \{\mathbf{x}'_i\}_{i=0}^{n'-1} \subset [0, 1)^s \right\},$$

defined in terms of a symmetric, positive definite kernel function $K : [0, 1)^{2s} \rightarrow \mathbb{C}$. The kernel allows us to define an inner product on $\mathcal{H}_{0,K}$ as

$$\langle f, g \rangle = \sum_{i=0}^{n_f-1} \sum_{j=0}^{n_g-1} \alpha_i \beta_j K(\mathbf{x}'_i, \mathbf{y}'_j)$$

for any two functions $f = \sum_{i=0}^{n_f-1} \alpha_i K(\cdot, \mathbf{x}'_i)$ and $g = \sum_{i=0}^{n_g-1} \beta_i K(\cdot, \mathbf{y}'_i)$. The linear space $\mathcal{H}_{0,K}$ may then be completed to obtain a Hilbert space, \mathcal{H}_K , for which K , is the reproducing kernel, see [1],

$$K(\cdot, \mathbf{y}) \in \mathcal{H}_K, \quad f(\mathbf{y}) = \langle K(\cdot, \mathbf{y}), f \rangle \quad \forall \mathbf{y} \in [0, 1)^s, \forall f \in \mathcal{H}_K.$$

The kernel functions considered here are based on Walsh functions and may be written as

$$K(\mathbf{x}, \mathbf{y}) = K(\mathbf{x} \ominus \mathbf{y}, \mathbf{0}) = \sum_{\mathbf{k} \in \mathbb{N}_0^s} \widehat{K}(\mathbf{k}) \text{wal}_{\mathbf{k}}(\mathbf{x} \ominus \mathbf{y}),$$

where positive definiteness is ensured by requiring $\widehat{K}(\mathbf{k})$ to be real and non-negative for all $\mathbf{k} \in \mathbb{N}_0^s$. The finiteness of this kernel is ensured by requiring that \widehat{K} be summable. The inner product for the Hilbert space defined by this kernel may be written as an ℓ_2 inner product in the spectral domain:

$$\langle f, g \rangle_{\mathcal{H}_K} = \sum_{\mathbf{k} \in \mathbb{N}_0^s} \frac{\widehat{f}(\mathbf{k}) \overline{\widehat{g}(\mathbf{k})}}{\widehat{K}(\mathbf{k})} = \left\langle \frac{\widehat{f}}{\sqrt{\widehat{K}}}, \frac{\widehat{g}}{\sqrt{\widehat{K}}} \right\rangle_2$$

where \widehat{f} and \widehat{g} are the Walsh coefficients of f and g . The accompanying norm is $\|f\|_{\mathcal{H}_K} = \langle f, f \rangle_{\mathcal{H}_K}^{1/2}$.

3.2. Interpolation of functions in the Walsh space. We now interpolate functions in \mathcal{H}_K using a linear combination of the reproducing kernel function where the second variable is fixed, $K(\cdot, \mathbf{x}_n)$, $n = 0, \dots, p^m - 1$, and where the \mathbf{x}_n are points taken from a digital net, $\mathcal{P}(\mathcal{C})$. Given the values of a function $f(\mathbf{x}_n)$, $n = 0, \dots, p^m - 1$, one can approximately recover it by a spline, defined as

$$Sf(\mathbf{x}) = \sum_{n=0}^{p^m-1} c_n K(\mathbf{x}, \mathbf{x}_n),$$

where the c_n are the coefficients to be found by interpolation: $f(\mathbf{x}_n) = Sf(\mathbf{x}_n)$ for $n = 0, \dots, p^m - 1$. This translates into solving the linear system

$$f(\mathbf{x}_n) = \sum_{v=0}^{p^m-1} c_v K(\mathbf{x}_n, \mathbf{x}_v) \quad \text{for } n = 0, \dots, p^m - 1,$$

for the coefficients c_0, \dots, c_{p^m-1} given the $f(\mathbf{x}_n)$ and $K(\mathbf{x}_n, \mathbf{x}_v)$. In the following paragraphs we show that the c_n can be computed in $\mathcal{O}(mp^m \log p)$ operations. See [30] for an analogue for lattice rules in the context of Fredholm integral equations of the second kind.

First observe that for any $\mathbf{k} \in \mathcal{K}(\mathcal{C})$ the DWT of the function data is

$$\begin{aligned} \widetilde{f}(\mathbf{k}) &= \frac{1}{p^m} \sum_{n=0}^{p^m-1} f(\mathbf{x}_n) \overline{\text{wal}_{\mathbf{k}}(\mathbf{x}_n)} = \frac{1}{p^m} \sum_{n=0}^{p^m-1} Sf(\mathbf{x}_n) \overline{\text{wal}_{\mathbf{k}}(\mathbf{x}_n)} \\ &= \frac{1}{p^m} \sum_{\mathbf{l} \in \mathbb{N}_0^s} \widehat{K}(\mathbf{l}) \sum_{r=0}^{p^m-1} c_r \overline{\text{wal}_{\mathbf{k}}(\mathbf{x}_r)} \sum_{n=0}^{p^m-1} \text{wal}_{\mathbf{l}}(\mathbf{x}_n) \overline{\text{wal}_{\mathbf{k}}(\mathbf{x}_n)} \\ (3.1) \quad &= p^m \sum_{\mathbf{d} \in \mathcal{D}(\mathcal{C})} \widehat{K}(\mathbf{k} \oplus \mathbf{d}) \widetilde{c}(\mathbf{k}) = p^m \widetilde{K}(\mathbf{k}) \widetilde{c}(\mathbf{k}), \end{aligned}$$

where $\widetilde{c}(\mathbf{k}) = p^{-m} \sum_{n=0}^{p^m-1} c_n \overline{\text{wal}_{\mathbf{k}}(\mathbf{x}_n)}$ is the DWT of the coefficients, $(c_n)_{n=0}^{p^m-1}$, and $\widetilde{K}(\mathbf{k}) = \sum_{\mathbf{l} \in \mathcal{D}(\mathcal{C})} \widehat{K}(\mathbf{k} \oplus \mathbf{l})$ is the DWT of the kernel data, $(K(\mathbf{x}_n, \mathbf{0}))_{n=0}^{p^m-1}$.

This implies that the DWT of the coefficients is essentially the quotient of the DWTs of the function data and the kernel data:

$$(3.2) \quad \tilde{c}(\mathbf{k}) = \frac{\tilde{f}(\mathbf{k})}{p^m \tilde{K}(\mathbf{k})}.$$

Using the FWT algorithm we can compute all $(\tilde{f}(\mathbf{k}))_{\mathbf{k} \in \mathcal{K}(\mathcal{C})}$ and $(\tilde{K}(\mathbf{k}))_{\mathbf{k} \in \mathcal{K}(\mathcal{C})}$, and hence $(\tilde{c}(\mathbf{k}))_{\mathbf{k} \in \mathcal{K}(\mathcal{C})}$ in $\mathcal{O}(mp^m \log p)$ operations. To compute c_0, \dots, c_{p^m-1} requires the inversion of the DWT, which is introduced in the next section.

3.3. Fast inverse discrete Walsh transform. The following lemma gives the key to the inverse discrete Walsh transform over a digital net.

Lemma 3.1. *Let $P(\mathcal{C}) = \{\mathbf{x}_0, \dots, \mathbf{x}_{p^m-1}\}$ be a digital net with p^m points. Let c_0, \dots, c_{p^m-1} be arbitrary complex numbers and let $(\tilde{c}(\mathbf{k}))_{\mathbf{k} \in \mathcal{K}(\mathcal{C})}$, denote the DWT of $(c_n)_{n=0}^{p^m-1}$. Then for $n = 0, \dots, p^m - 1$ the coefficients are*

$$c_n = \sum_{\mathbf{k} \in \mathcal{K}(\mathcal{C})} \tilde{c}(\mathbf{k}) \text{wal}_{\mathbf{k}}(\mathbf{x}_n).$$

Proof. The sum on the right of the above equation is

$$\begin{aligned} \sum_{\mathbf{k} \in \mathcal{K}(\mathcal{C})} \tilde{c}(\mathbf{k}) \text{wal}_{\mathbf{k}}(\mathbf{x}_n) &= \sum_{v=0}^{p^m-1} c_v \frac{1}{p^m} \sum_{\mathbf{k} \in \mathcal{K}(\mathcal{C})} \overline{\text{wal}_{\mathbf{k}}(\mathbf{x}_v)} \text{wal}_{\mathbf{k}}(\mathbf{x}_n) \\ &= \sum_{v=0}^{p^m-1} c_v \frac{1}{p^m} \sum_{\mathbf{k} \in \mathcal{K}(\mathcal{C})} \text{wal}_{\mathbf{k}}(\mathbf{x}_n \ominus \mathbf{x}_v). \end{aligned}$$

The definition of the Walsh function and the net imply that $\text{wal}_{\mathbf{k}}(\mathbf{x}_n \ominus \mathbf{x}_v) = \text{wal}_{\mathbf{k}}(\mathbf{x}_{n \ominus v}) = \omega_p^{(\vec{n} \ominus \vec{v}) \cdot \vec{h}}$, where $\mathbf{k} \in \mathcal{D}(\vec{h}) \cap \mathcal{K}(\mathcal{C})$. Thus we have

$$\frac{1}{p^m} \sum_{\mathbf{k} \in \mathcal{K}(\mathcal{C})} \text{wal}_{\mathbf{k}}(\mathbf{x}_n \ominus \mathbf{x}_v) = \frac{1}{p^m} \sum_{\vec{h} \in \mathbb{Z}_p^m} \omega_p^{(\vec{n} \ominus \vec{v}) \cdot \vec{h}}$$

and the last sum is 1 if $\vec{n} = \vec{v}$, i.e. $n = v$, and 0 otherwise. Hence the result follows. \square

The above lemma describes the inversion of the discrete Walsh transform which we might call the inverse discrete Walsh transform. As for the discrete Walsh transform, there is also a fast inversion of the discrete Walsh transform which we describe in the following.

Algorithm 2 (Fast Inverse Discrete Walsh Transform (FIWT)). For $(h_0, \dots, h_{m-1} \in \mathbb{Z}_p)$ we define $D^{(0)}(h_0, \dots, h_{m-1}) = \tilde{c}(\mathbf{k})$ where $\mathbf{k} \in \mathcal{D}(\vec{h}) \cap \mathcal{K}(\mathcal{C})$. Then for $r = 1, 2, \dots, m$ compute for all $h_r, \dots, h_{m-1} \in \mathbb{Z}_p$ and all $n_0, \dots, n_{r-1} \in \mathbb{Z}_p$ the sums

$$\begin{aligned} D^{(r)}(n_0, \dots, n_{r-1}, h_r, \dots, h_{m-1}) \\ = \sum_{h_{r-1}=0}^{p-1} \omega_p^{h_{r-1} n_{r-1}} D^{(r-1)}(n_0, \dots, n_{r-2}, h_{r-1}, h_r, \dots, h_{m-1}). \end{aligned}$$

Then for $n = 0, \dots, p^m - 1$ with $n = n_0 + \dots + n_{m-1} p^{m-1}$ let

$$c_n = D^{(m)}(n_0, \dots, n_{m-1}).$$

Hence c_0, \dots, c_{p^m-1} can also be computed from $(\tilde{c}(\mathbf{k}))_{\mathbf{k} \in \mathcal{K}(\mathcal{C})}$ in $\mathcal{O}(mp^{m+1})$ operations. As with the FWT algorithm, this number of operations can be further reduced to $\mathcal{O}(mp^m \log p)$ if we apply Rader's algorithm. Thus the spline interpolation at a point, $Sf(\mathbf{x})$, can be computed in $\mathcal{O}(mp^m \log p)$ operations, and each additional point can be computed in $\mathcal{O}(p^m)$ operations.

3.4. Best possible interpolation. Splines as defined above provide optimal interpolation of a function in at least two senses. The spline approximation is the smallest function in \mathcal{H}_K that interpolates the data:

$$Sf = \min_{\substack{g \in \mathcal{H}_K \\ g(\mathbf{x}_i) = f(\mathbf{x}_i), \ i=0, \dots, p^m}} \|g\|_{\mathcal{H}_K}.$$

Moreover, the spline algorithm is the best linear algorithm, i.e., algorithm of the form $Af = \sum_{i=0}^{n-1} f(\mathbf{x}_i)w_i(\mathbf{x})$ for any choice of the $w_i(\mathbf{x})$:

$$Sf(\mathbf{x}) = \operatorname{argmin}_{Af} \sup_{\|f\|_{\mathcal{H}(K)} \leq 1} |f(\mathbf{x}) - Af(\mathbf{x})|.$$

The proofs of these assertions are contained in [8, Chapter 18] and elsewhere.

4. ANOVA DECOMPOSITION OF THE INTERPOLATION

Based on the spline interpolation of a function, one can also approximate its analysis of variance (ANOVA) effects, i.e., the pieces of the function depending on a subset of the s variables. This provides a way to estimate the truncation and superposition dimensions of functions via their spline interpolations.

Let $1:s$ denote the set of coordinate indices, $\{1, \dots, s\}$, for short. Let u denote a subset of $1:s$ and let \bar{u} denote $1:s \setminus u$. For any $\mathbf{x} \in [0, 1]^s$, let $\mathbf{x}_u = (x_j)_{j \in u}$ denote the vector of coordinates indexed by u . The ANOVA decomposition [7, 22] of a function $f : [0, 1]^s \rightarrow \mathbb{R}$, is denoted

$$(4.1a) \quad f(\mathbf{x}) = \sum_{u \subseteq 1:s} f_u(\mathbf{x}_u),$$

where the ANOVA effect, f_u , is defined recursively by taking the integral over $[0, 1]^{\bar{u}}$ and then subtracting the lower order effects:

$$(4.1b) \quad f_\emptyset = \int_{[0,1]^s} f(\mathbf{x}) d\mathbf{x}, \quad f_u(\mathbf{x}_u) = \int_{[0,1]^{\bar{u}}} f(\mathbf{x}) d\mathbf{x}_{\bar{u}} - \sum_{v \subset u} f_v(\mathbf{x}_v),$$

We emphasize that \subset on the right side of this last equation denotes the proper subset. Also, $[0, 1]^u$ denotes the Cartesian product of $|u|$ copies of $[0, 1]$, where $|u|$ is the cardinality of u .

The ANOVA effects, f_u , of the function $f \in \mathcal{H}_K$ lie in subspaces, \mathcal{H}_{K_u} for kernels constructed appropriately. The kernels K_u are products of a univariate kernel, K' .

The kernel K' for univariate functions is defined as

$$K'(x, y) = K'(x \ominus y, 0) = \sum_{k=1}^{\infty} \widehat{K'}(k) \operatorname{wal}_k(x \ominus y),$$

where the Walsh coefficients of the kernel must be non-negative. One reasonable choice is

$$\widehat{K'}(k) = \frac{p^\alpha - p}{p^\alpha(p-1)} p^{-\alpha a}$$

for $k = k_0 + k_1p + \cdots + k_ap^a \in \mathbb{N}$ with $k_a \neq 0$, and where $\alpha > 1$ is a parameter that measures the digital smoothness of the kernel. These Walsh coefficients have been normalized so that

$$K'(x, x) = K'(0, 0) = \sum_{k=1}^{\infty} \widehat{K'}(k) = 1.$$

Because K' does not include the constant function, i.e., $\widehat{K'}(0) = 0$ implicitly, it follows that $\int_0^1 K'(x, y) dy = 0$. A computable short form of K' can be obtained [6], namely for $x = x_1p^{-1} + x_2p^{-2} + \cdots$ and $y = x_1p^{-1} + x_2p^{-2} + \cdots + x_{i-1}p^{-i+1} + y_ip^{-i} + y_{i+1}p^{-i-1} + \cdots$ with $y_i \neq x_i$ we have

$$K'(x, y) = K'(x \ominus y, 0) = 1 - p^{i(1-\alpha)} \frac{p^\alpha - 1}{p - 1}.$$

The kernel for functions of s variables is a product involving K' , namely,

$$\begin{aligned} K(\mathbf{x}, \mathbf{y}) &= K(\mathbf{x} \ominus \mathbf{y}, \mathbf{0}) = \sum_{\mathbf{k} \in \mathbb{N}_0^s} \widehat{K}(\mathbf{k}) \text{wal}_{\mathbf{k}}(\mathbf{x} \ominus \mathbf{y}) \\ &= \prod_{j=1}^s [1 + \gamma_j K'(x_j, y_j)] = \prod_{j=1}^s [1 + \gamma_j K'(x_j \ominus y_j, 0)] \\ &= \sum_{u \subseteq 1:s} \gamma_u K_u(\mathbf{x}_u, \mathbf{y}_u) = \sum_{u \subseteq 1:s} \gamma_u K_u(\mathbf{x}_u \ominus \mathbf{y}_u, \mathbf{0}), \end{aligned}$$

where the tuning parameter γ_u has a product form,

$$\gamma_u = \prod_{j \in u} \gamma_j, \quad \gamma_\emptyset = 1,$$

and the Walsh coefficients of these multivariate kernels have expressions in terms of the Walsh coefficients of K' :

$$\begin{aligned} \widehat{K}(\mathbf{k}) &= \prod_{j=1}^s [\delta_{k_j, 0} + \gamma_j \widehat{K'}(k_j)] = \sum_{u \subseteq 1:s} \gamma_u \widehat{K}_u(\mathbf{k}_u) \delta_{\mathbf{k}_{\bar{u}}, \mathbf{0}}, \\ K_u(\mathbf{x}_u, \mathbf{y}_u) &= K_u(\mathbf{x}_u \ominus \mathbf{y}_u, \mathbf{0}) = \sum_{\mathbf{k}_u \in \mathbb{N}_0^u} \widehat{K}_u(\mathbf{k}_u) \text{wal}_{\mathbf{k}_u}(\mathbf{x} \ominus \mathbf{y}) \\ &= \prod_{j \in u} K'(x_j, y_j) = \prod_{j \in u} K'(x_j \ominus y_j, 0), \\ \widehat{K}_u(\mathbf{k}_u) &= \prod_{j \in u} \widehat{K'}(k_j), \quad \widehat{K}_\emptyset = 1. \end{aligned}$$

The spline approximation via this kernel may be expressed as the sum of its ANOVA effects:

$$Sf(\mathbf{x}) = \sum_{n=0}^{p^m-1} c_n K(\mathbf{x}, \mathbf{x}_n) = \sum_{n=0}^{p^m-1} c_n K(\mathbf{x} \ominus \mathbf{x}_n, \mathbf{0}) = \sum_{u \subseteq 1:s} (Sf)_u(\mathbf{x}_u),$$

where these ANOVA effects are written in terms of the kernels K_u :

$$(4.2) \quad (Sf)_u(\mathbf{x}_u) = \gamma_u \sum_{n=0}^{p^m-1} c_n K_u(\mathbf{x}_u, \mathbf{x}_{n,u}) = \gamma_u \sum_{n=0}^{p^m-1} c_n K_u(\mathbf{x}_u \ominus \mathbf{x}_{n,u}, \mathbf{0}).$$

These $(Sf)_u$ may be verified as the ANOVA effects of Sf defined in (4.1) by noting that $\int_0^1 K'(x, y) dy = 0$ for all x .

The special form of the reproducing kernel defined here facilitates the calculation of the variance of $(Sf)_u$, denoted $\sigma^2((Sf)_u)$. Noting that for $|u| > 0$, $(Sf)_u$ has zero mean, it follows that

$$\begin{aligned} \sigma^2((Sf)_u) &= \int_{[0,1]^u} \left[Sf_u(\mathbf{x}_u) - \int_{[0,1]^u} Sf_u(\mathbf{x}'_u) d\mathbf{x}'_u \right]^2 d\mathbf{x}_u \\ &= \int_{[0,1]^u} [Sf_u(\mathbf{x}_u)]^2 d\mathbf{x}_u \\ &= \gamma_u^2 \sum_{n,v=0}^{p^m-1} c_n c_v \int_{[0,1]^u} K_u(\mathbf{x}_u, \mathbf{x}_{n,u}) K_u(\mathbf{x}_u, \mathbf{x}_{v,u}) d\mathbf{x}_u \\ &= \gamma_u^2 \sum_{n,v=0}^{p^m-1} c_n c_v \prod_{j \in u} \int_0^1 K'(x_j, x_{n,j}) K'(x_j, x_{v,j}) dx_j. \end{aligned}$$

Substituting the Walsh expansions of the univariate kernels in the above integral and noting that the Walsh functions are orthogonal yields an expression for the integral in terms of a related univariate kernel:

$$\int_0^1 K'(x_j, x_{n,j}) K'(x_j, x_{v,j}) dx_j = R'(x_{n,j}, x_{v,j}),$$

where

$$\begin{aligned} R'(x, y) &= R'(x \ominus y, 0) = \sum_{k=0}^{\infty} \widehat{R}'(k) \text{wal}_k(x \ominus y), \quad \widehat{R}'(k) = \left[\widehat{K}'(k) \right]^2, \\ \widehat{R}'(0) &= 0 \quad \text{and} \quad \widehat{R}'(k) = \left[\frac{p^\alpha - p}{p^\alpha(p-1)} \right]^2 p^{-2\alpha a} \end{aligned}$$

for $k = k_0 + k_1 p + \dots + k_a p^a \in \mathbb{N}$ with $k_a \neq 0$. Moreover, as is the case for K' , there is a computationally simple form for R' as well. For $x = x_1 p^{-1} + x_2 p^{-2} + \dots$ and $y = x_1 p^{-1} + x_2 p^{-2} + \dots + x_{i-1} p^{-i+1} + y_i p^{-i} + y_{i+1} p^{-i-1} + \dots$ with $y_i \neq x_i$ we have

$$R'(x, y) = R'(x \ominus y, 0) = \frac{(p^\alpha - p)^2}{(p-1)(p^{2\alpha} - p)} \left[1 - p^{i(1-2\alpha)} \frac{p^{2\alpha} - 1}{p-1} \right].$$

The product of the univariate kernels R' is then used to define the kernels R_u :

$$\begin{aligned} R_u(\mathbf{x}_u, \mathbf{y}_u) &= R_u(\mathbf{x}_u \ominus \mathbf{y}_u, \mathbf{0}) = \prod_{j \in u} R'(x_j, y_j) = \sum_{\mathbf{k} \in \mathbb{N}_0^u} \widehat{R}_u(\mathbf{k}_u) \text{wal}_{\mathbf{k}_u}(\mathbf{x}_u \ominus \mathbf{y}_u), \\ \widehat{R}_u(\mathbf{k}_u) &= \prod_{j \in u} \widehat{R}'(k_j). \end{aligned}$$

This definition allows the variance of $(Sf)_u$ to be written in terms of the kernel R_u :

$$\sigma^2((Sf)_u) = \gamma_u^2 \sum_{n,v=0}^{p^m-1} c_n c_v R_u(\mathbf{x}_{n,u}, \mathbf{x}_{v,u}).$$

The sum above may be evaluated naively using $\mathcal{O}(p^{2m})$ operations. However, using the FWT allows evaluation with only $\mathcal{O}(mp^m \log p)$ operations. First, the

spline coefficients, c_n and c_v are written in terms of their inverse discrete Walsh transform coefficients via Lemma 3.1:

$$\begin{aligned}\sigma^2((Sf)_u) &= \gamma_u^2 \sum_{n,v=0}^{p^m-1} \sum_{\mathbf{k} \in \mathcal{K}(\mathcal{C})} \tilde{c}(\mathbf{k}) \text{wal}_{\mathbf{k}}(\mathbf{x}_n) \sum_{\mathbf{l} \in \mathcal{K}(\mathcal{C})} \overline{\tilde{c}(\mathbf{l}) \text{wal}_{\mathbf{l}}(\mathbf{x}_v)} R_u(\mathbf{x}_{n,u}, \mathbf{x}_{v,u}) \\ &= \gamma_u^2 \sum_{\mathbf{k}, \mathbf{l} \in \mathcal{K}(\mathcal{C})} \tilde{c}(\mathbf{k}) \overline{\tilde{c}(\mathbf{l})} \sum_{n,v=0}^{p^m-1} R_u(\mathbf{x}_{n,u} \ominus \mathbf{x}_{v,u}, \mathbf{0}) \text{wal}_{\mathbf{k}}(\mathbf{x}_n) \overline{\text{wal}_{\mathbf{l}}(\mathbf{x}_v)}.\end{aligned}$$

The double sum of the kernel R_u may be further simplified by applying certain elementary properties of Walsh functions for any $\mathbf{k}, \mathbf{l} \in \mathcal{K}(\mathcal{C})$:

$$\begin{aligned}\sum_{n,v=0}^{p^m-1} R_u(\mathbf{x}_{n,u} \ominus \mathbf{x}_{v,u}, \mathbf{0}) \text{wal}_{\mathbf{k}}(\mathbf{x}_n) \overline{\text{wal}_{\mathbf{l}}(\mathbf{x}_v)} \\ &= \sum_{n,v=0}^{p^m-1} R_u(\mathbf{x}_{n,u} \ominus \mathbf{x}_{v,u}, \mathbf{0}) \text{wal}_{\mathbf{k}}(\mathbf{x}_n \ominus \mathbf{x}_{v,u}) \text{wal}_{\mathbf{k} \ominus \mathbf{l}}(\mathbf{x}_v) \\ &= \sum_{\mathbf{x} \in P(\mathcal{C})} R_u(\mathbf{x}_u, \mathbf{0}) \text{wal}_{\mathbf{k}}(\mathbf{x}) \sum_{n,v=0}^{p^m-1} \text{wal}_{\mathbf{k} \ominus \mathbf{l}}(\mathbf{x}_v) = p^{2m} \delta_{\mathbf{k}, \mathbf{l}} \widetilde{R}_u(\mathbf{k})\end{aligned}$$

where the DWT of R_u is

$$\widetilde{R}_u(\mathbf{k}) = \frac{1}{p^m} \sum_{\mathbf{x} \in P(\mathcal{C})} R_u(\mathbf{x}_u, \mathbf{0}) \text{wal}_{\mathbf{k}}(\mathbf{x}).$$

Substituting the formula for the double sum of the kernel R_u in terms of its DWT yields

$$(4.3) \quad \sigma^2((Sf)_u) = \gamma_u^2 p^{2m} \sum_{\mathbf{k} \in \mathcal{K}(\mathcal{C})} |\tilde{c}(\mathbf{k})|^2 \widetilde{R}_u(\mathbf{k}).$$

Since the DWT of the spline coefficients and R_u may each be calculated in $\mathcal{O}(mp^m \log p)$ operations by the FWT algorithm, it follows that each $\sigma^2((Sf)_u)$ may be calculated in $\mathcal{O}(mp^m \log p)$ operations.

Calculating the variances of all the Sf_u may be too burdensome, since there are 2^s ANOVA effects. However, sums of the $\sigma^2((Sf)_u)$ are useful for determining the effective dimension of f [3], which is an important factor in the performance of quasi-Monte Carlo methods.

The truncation variance of order d , denoted $\sigma_{\text{trc}}^2(f; d)$ is defined as sum of the variances of all ANOVA effects involving the first d or fewer variables:

$$\sigma_{\text{trc}}^2(f; d) = \sum_{u \subseteq 1:d} \sigma^2(f_u).$$

The superposition variance of order d , denoted $\sigma_{\text{sup}}^2(f; d)$ is defined as sum of the variances of all ANOVA effects involving d or fewer variables:

$$\sigma_{\text{sup}}^2(f; d) = \sum_{0 < |u| \leq d} \sigma^2(f_u).$$

It follows from these two definitions that the truncation variance is no more than the superposition variance, and for all d between 0 and s ,

$$0 = \sigma_{\text{trc}}^2(f; 0) = \sigma_{\text{sup}}^2(f; 0) \leq \sigma_{\text{trc}}^2(f; d) \leq \sigma_{\text{sup}}^2(f; d) \leq \sigma_{\text{trc}}^2(f; s) = \sigma_{\text{sup}}^2(f; s) = \sigma^2(f).$$

The truncation dimension, d_{trc} , and the superposition dimension, d_{sup} are defined as the smallest dimensions for which the truncation and superposition variances, respectively, are 99% of the total variance of the function, i.e.,

$$\begin{aligned} \sigma_{\text{trc}}^2(f; d_{\text{trc}} - 1) &< 0.99\sigma^2(f) \leq \sigma_{\text{trc}}^2(f; d_{\text{trc}}), \\ \sigma_{\text{trc}}^2(f; d_{\text{sup}} - 1) &< 0.99\sigma^2(f) \leq \sigma_{\text{sup}}^2(f; d_{\text{sup}}). \end{aligned}$$

The truncation and superposition dimensions of a function may be estimated by the truncation and superposition dimensions of their spline approximations. To do this requires computationally efficient formulas for the truncation and superposition variances. The truncation variance may be written as

$$\begin{aligned} \sigma_{\text{trc}}^2(Sf; d) &= \sum_{u \subseteq 1:d} \sigma^2((Sf)_u) = \sum_{u \subseteq 1:d} \gamma_u^2 p^{2m} \sum_{\mathbf{k} \in \mathcal{K}(\mathcal{C})} |\tilde{c}(\mathbf{k})|^2 \widetilde{R}_u(\mathbf{k}) \\ &= p^{2m} \sum_{\mathbf{k} \in \mathcal{K}(\mathcal{C})} |\tilde{c}(\mathbf{k})|^2 \widetilde{R}_d(\mathbf{k}), \end{aligned}$$

where \widetilde{R}_d is the DWT of the kernel R_d defined as

$$\begin{aligned} R_d(\mathbf{x}, \mathbf{y}) &= R_d(\mathbf{x} \ominus \mathbf{y}, \mathbf{0}) \\ &= \sum_{u \subseteq 1:d} \gamma_u^2 R_u(\mathbf{x}_u, \mathbf{y}_u) = \sum_{u \subseteq 1:d} \gamma_u^2 R_u(\mathbf{x}_u \ominus \mathbf{y}_u, \mathbf{0}) \\ &= \prod_{j=1}^d [1 + \gamma_j^2 R'(x_j, y_j)] = \prod_{j=1}^d [1 + \gamma_j^2 R'(x_j \ominus y_j, 0)]. \end{aligned}$$

By convention $R(\mathbf{x}, \mathbf{y})$ is defined as $R_s(\mathbf{x}, \mathbf{y})$. Note that the total variance of the spline approximation to f is given by taking $d = s$ above:

$$(4.4) \quad \sigma^2(Sf) = p^{2m} \sum_{\mathbf{k} \in \mathcal{K}(\mathcal{C})} |\tilde{c}(\mathbf{k})|^2 \widetilde{R}(\mathbf{k}) = p^{2m} \sum_{\mathbf{k} \in \mathcal{K}(\mathcal{C})} |\tilde{f}(\mathbf{k})|^2 \frac{\widetilde{R}(\mathbf{k})}{|\widetilde{K}(\mathbf{k})|^2},$$

where (3.2) has been used.

For γ_j of the form $\beta \tilde{\gamma}_j$ and thus $\gamma_u = \beta^{|u|} \tilde{\gamma}_u$ the DWT $\widetilde{R}(\mathbf{k})$ may be written as an s -degree polynomial in β^2 with vanishing constant term:

$$\begin{aligned} R(\mathbf{x}, \mathbf{y}) &= R(\mathbf{x} \ominus \mathbf{y}, \mathbf{0}) = \sum_{j=1}^s \beta^{2j} \sum_{\substack{u \subseteq 1:s \\ |u|=j}} \tilde{\gamma}_u^2 R_u(\mathbf{x}_u \ominus \mathbf{y}_u, \mathbf{0}), \\ \widetilde{R}(\mathbf{k}) &= \sum_{j=1}^s \beta^{2j} \widetilde{Q}(\mathbf{k}, j), \end{aligned}$$

where the $\tilde{Q}(\mathbf{k}, j)$ are the coefficients of this polynomial in β^2 and are given by

$$\tilde{Q}(\mathbf{k}, j) = \tilde{\gamma}_u^2 \sum_{\substack{u \subseteq 1:s \\ |u|=j}} \tilde{R}_u(\mathbf{k}).$$

The coefficients $\tilde{Q}(\mathbf{k}, j)$ may be obtained by evaluating $\tilde{R}(\mathbf{k})$ at s different values of β^2 , and then performing polynomial interpolation, a relatively inexpensive procedure requiring $\mathcal{O}(s^2)$ operations. These $\tilde{Q}(\mathbf{k}, j)$ may then be used to evaluate the superposition variance as follows:

$$\begin{aligned} \sigma_{\text{sup}}^2(f; d) &= \sum_{0 < |u| \leq d} \sigma^2((Sf)_u) = \sum_{0 < |u| \leq d} \gamma_u^2 p^{2m} \sum_{\mathbf{k} \in \mathcal{K}(\mathcal{C})} |\tilde{c}(\mathbf{k})|^2 \tilde{R}_u(\mathbf{k}) \\ &= p^{2m} \sum_{\mathbf{k} \in \mathcal{K}(\mathcal{C})} |\tilde{c}(\mathbf{k})|^2 \sum_{0 < |u| \leq d} \beta^{2|u|} \tilde{\gamma}_u^2 \tilde{R}_u(\mathbf{k}) \\ &= p^{2m} \sum_{\mathbf{k} \in \mathcal{K}(\mathcal{C})} |\tilde{c}(\mathbf{k})|^2 \sum_{j=1}^d \beta^{2j} \tilde{Q}(\mathbf{k}, j) \\ &= \sigma_{\text{sup}}^2(f; d-1) + p^{2m} \beta^{2d} \sum_{\mathbf{k} \in \mathcal{K}(\mathcal{C})} |\tilde{c}(\mathbf{k})|^2 \tilde{Q}(\mathbf{k}, d). \end{aligned}$$

Thus, both the truncation and superposition dimensions may be evaluated via the FWT algorithm using $\mathcal{O}(s^2 m p^m \log(p))$ operations.

5. NUMERICAL RESULTS

5.1. Optimization of the Kernel Parameters. The γ_j are weights in the ANOVA decomposition of the kernel and α controls the rate of decay of Fourier coefficients of the kernel. Their optimal values for spline interpolation depend on the particular function f to be interpolated. Assume that $p = 2$, and assume that we have a digital net with $2N$ points, $\{\mathbf{x}_0, \dots, \mathbf{x}_{2N-1}\}$, whose first N points themselves constitute a digital net, as do the second N points. We can construct the spline interpolant by the values of $f(\mathbf{x}_n)$ for $n = 0, \dots, N-1$, and then estimate the error of the spline by the cost function

$$(5.1) \quad \sum_{n=N+1}^{2N-1} (f(\mathbf{x}_n) - Sf(\mathbf{x}_n))^2.$$

The values of $Sf(\mathbf{x}_n)$ for $n = N, \dots, 2N-1$ can be evaluated by a fast discrete Walsh transform. There are s optimization parameters γ_j and the optimization process becomes slow when s increases. In order to reduce the number of optimization parameters, the values of γ_j are assumed to be of the following form in this section,

$$\gamma_j = \beta j^q$$

where q is a new optimization parameter. Thus, for any s , there are only three optimization parameters: β , α and q . Their optimal values in Sections 5.2 and 5.3 were found by optimizing (5.1) using the function `fminsearch` in MATLAB.

a_k	s	d_{trc}			d_{sup}		
		Exact	Wang & Fang	Spline	Exact	Wang & Fang	Spline
1	10	10	10	10	3	-	2
	20	20	20	20	5	-	2
	40	40	39	40	8	-	2
k	10	10	10	10	2	-	2
	20	18	19	18	2	-	2
	40	33	34	31	2	-	2
k^2	10	5	5	5	2	-	2
	20	5	5	5	2	-	2
	40	5	5	5	2	-	2

TABLE 1. The effective dimensions of the multiplicative functions

5.2. Effective Dimension of Multiplicative Functions. Consider a class of test functions in [25]

$$f(\mathbf{x}) = \prod_{k=1}^s \frac{|4x_k - 2| + a_k}{1 + a_k},$$

where a_k are parameters. We consider three possible choices of a_k : $a_k = 1$, $a_k = k$ and $a_k = k^2$ for $k = 1 \dots s$. The effective dimension in both senses of these functions can be computed analytically, or estimated by the algorithm in [25]. We compare them with the effective dimensions of the interpolating spline as described in the previous section. Table 1 shows the results using a Sobol point set with $m = 12$ and $p = 2$, so $N = 4096$.

Note that d_{sup} cannot be calculated by the method in [25]. The spline method described in the previous section can estimate both d_{sup} and d_{trc} well in all cases except for d_{sup} in the case $a_k = 1$ and $s = 20, 40$. Table 2 shows $\sigma^2(f)$ by analytical calculation and two approximations: $\sigma_{\text{QMC}}^2(f) = \frac{1}{N} \sum_{n=0}^{N-1} f^2(\mathbf{x}_n) - (\frac{1}{N} \sum_{n=0}^{N-1} f(\mathbf{x}_n))^2$ and $\sigma^2(Sf)$ by (4.4). Table 2 also shows the ratio $\sigma^2(Sf)/\sigma_{\text{QMC}}^2(f)$ as an indicator of how well these two approximations agree. One can prove that $\sigma^2(Sf)/\sigma_{\text{QMC}}^2(f) \leq 1$. It is found that the values of $\sigma^2(Sf)/\sigma_{\text{QMC}}^2(f)$ are close to unity for most cases but are extremely small for $a_k = 1, s = 20, 40$. It was reported in [26] that the errors of the numerical integration by the QMC methods were also large for these latter cases.

5.3. Asian Option Pricing. The pricing of an Asian option is based on the arithmetic average of the stock prices in a particular period of time. The payoff of the call option at the end of period T is

$$\text{payoff} = \max \left(\frac{1}{s} \sum_{j=1}^s S_j - K, 0 \right),$$

where S_j is the stock price at time $t_j = jT/s, j = 0, \dots, s, t_s = T$ and K is the strike price. Based on the risk-neutral valuation principle, the price of the call option at $t = 0$ should be

$$E(e^{-rT} \text{payoff}),$$

a_k	s	$\sigma^2(f)$	$\sigma_{\text{QMC}}^2(f)$	$\sigma^2(Sf)$	$\sigma^2(Sf)/\sigma_{\text{QMC}}^2(f)$
1	10	1.2265	1.9990	1.0434	0.5220
	20	3.9573	2.7×10^3	0.2692	9.9704×10^{-5}
	40	23.5745	2.98×10^{10}	0.2393	8.0302×10^{-12}
k	10	0.1992	0.2025	0.1960	0.9679
	20	0.2154	0.2340	0.2073	0.8859
	40	0.2246	0.3134	0.2088	0.6662
k^2	10	0.1038	0.1039	0.1037	0.9981
	20	0.1039	0.1040	0.1038	0.9981
	40	0.1039	0.1041	0.1038	0.9971

TABLE 2. The variances of the multiplicative functions estimated by a QMC method and the variances of the splines.

s	Wang & Fang			Spline	
	standard	BB	PCA	d_{trc}	d_{sup}
8	7	5	2	7	2
16	14	7	2	14	2
32	27	7	2	27	2

TABLE 3. The estimated effective dimensions of the Asian option pricing problem.

where $E(\cdot)$ is the expected value of different path movements of the stock price and r is the risk-free interest rate. The stock price movement is assumed to follow a geometric Brownian motion,

$$S_j = S_{j-1} e^{(r-0.5\sigma^2)T/s + \sigma\sqrt{T/s}Z_j},$$

where σ is the volatility, and Z_1, \dots, Z_s are independent standard normal random variables. See [27] for the details.

Table 3 shows the results of the estimated effective dimension in both senses for an Asian option pricing problem with the following parameters: $N = 2^{14}$, $S_0 = K = 100$, $\sigma = 0.2$, $r = 0.1$, and $T = 1$ year using a Sobol point set. The results in [25] are also duplicated. The column “standard” is the estimated truncation dimension of the original pricing function by their method. The columns “BB” and “PCA” are the estimated truncation dimensions when the Brownian bridge and principle component analysis dimension reduction methods are applied, respectively. The d_{trc} of the spline are exactly the same as the results in [25]. The d_{sup} are also the same as the truncation dimension after applying PCA, which is the best dimension reduction method in [25]. Moreover, the superposition dimension $d_{\text{sup}} = 2$ is consistent with the analytical results in [27]. The variances of the discounted payoff for the Asian option pricing problem computed by the sample variance and the variance of the spline are shown in Table 4, which indicate that the accuracy of the spline for estimating effective dimension should be reasonably good.

s	$\sigma_{\text{QMC}}^2(f)$	$\sigma^2(Sf)$	$\sigma^2(Sf)/\sigma_{\text{QMC}}^2(f)$
8	70.3669	69.4148	0.9865
16	71.6402	69.0217	0.9634
32	72.6395	67.8101	0.9335

TABLE 4. The variances of the Asian option price problem estimated by a QMC method and the variances of the splines.

6. CONCLUSION AND REMARKS

A fast discrete Walsh transform over digital nets is derived in this article. This transform can be applied to reduce the cost of calculating the discrete Walsh coefficients from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log N)$. Because the kernel used is piecewise constant, the spline interpolant based on this kernel may not be particularly accurate. However, the spline interpolant does facilitate the estimation of coarser quantities, such as the variances of the ANOVA effects and the effective dimension of the function. The numerical results for two different families of functions are compared with the analytical results and the results from the literature showing that the estimation of effective dimension via the spline is accurate and efficient.

Finally, we would like add two remarks. The function values in our interpolation method were sampled over a digital net, which was originally constructed for numerical quadrature. Sparse grids [2] are another sampling scheme and approximation method that can be applied to both numerical quadrature and function interpolation. Both methods can be classified as *a priori* grid optimization [2]. However, the point sets are based on different criteria. Digital nets minimize the discrepancy of the point set [14], while the sparse grid is selected based on the importance of the components in a tensor product of hierarchical function spaces. Sparse grids use $\mathcal{O}(h^{-1}(\log h^{-1})^{s-1})$ points where h is the mesh size in each dimension. This number is substantially smaller than $\mathcal{O}(h^{-s})$, the number of points needed for an ordinary grid, but the number of points for a sparse grid increases exponentially with dimension for a given mesh size. On the other hand there exist digital nets with $N = p^m$ points for $m = 0, 1, \dots$, independent of the dimension, s . Error bounds have been developed for sparse grid methods. An error analysis of the spline algorithm for digital nets is the object of future work.

The second remark is that the fast algorithms in this paper are due to the correspondence between the point set and the kernel. In [29], a similar technique was applied to integration lattices and the fast Fourier transform (FFT). For computer experiments, one typically has control over where to sample the underlying function, but for some problems of approximating a function based on observational or field data, the locations of the points where the function is sampled can not be selected in advance. Such problems are suited to the Nonequidistant Fast Fourier Transform (NFFT) pioneered by Potts [17]. The NFFT provides a fast and relatively accurate, but only approximate, discrete Fourier transform of the data. The FFT applied to data sampled on an integration lattice and the Fast Walsh Transform (FWT) introduced here applied to data sampled on digital nets both provide fast discrete transforms with accuracy only limited by machine precision.

REFERENCES

1. N. Aronszajn, Theory of reproducing kernels. *Trans. Amer. Math. Soc.*, 68, 337–404, 1950.
2. H. Bungartz and M. Griebel, Sparse grids. *Acta Numer.*, 13, 147–269, 2004.
3. R. Caflisch, R. Morokoff and A. Owen, Valuation of mortgage backed securities using Brownian bridges to reduce the effective dimension. *J. Comput. Finance*, 1, 27–46, 1997.
4. H.E. Chrestenson, A class of generalized Walsh functions. *Pacific J. Math.*, 5, 17–31, 1955.
5. J.W. Cooley and J.W. Tukey, An algorithm for the machine calculation of complex Fourier series. *Math. Comp.*, 19, 297–301, 1965.
6. J. Dick and F. Pillichshammer, Multivariate integration in weighted Hilbert spaces based on Walsh functions and weighted Sobolev spaces. *J. Complexity*, 21, 149–195, 2005.
7. B. Efron and C. Stein, The Jackknife estimate of variance. *Ann. Statist.*, 9, 586–596, 1981.
8. G. E. Fasshauer, *Meshfree Approximation Methods with Matlab*, Interdisciplinary Mathematical Sciences Series, vol. 6, World Scientific Publishing Co., Singapore, 2007.
9. H. Faure, Discrepance de suites associées à un système de numération (en dimension s). *Acta Arith.*, 41, 337–351, 1982.
10. G. Larcher, A class of low-discrepancy point-sets and its application to numerical integration by number-theoretical methods. *Grazer Math. Ber.*, 318, 69–80, 1993.
11. G. Larcher and C. Traunfellner, On the numerical integration of Walsh series by number-theoretic methods. *Math. Comp.*, 63, 277–291, 1994.
12. D. Li and F.J. Hickernell, Trigonometric spectral collocation methods on lattices. *Contemp. Math.*, 330, 121–132, 2003.
13. H. Niederreiter, Low-discrepancy point sets. *Monatsh. Math.*, 102, 155–167, 1986.
14. H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*. No. 63 in CBMS-NSF Series in Applied Mathematics. SIAM, Philadelphia, 1992.
15. H. Niederreiter, Constructions of (t, m, s) -nets and (t, s) -sequences. *Finite Fields Appl.*, 11, 578–600, 2005.
16. H. Niederreiter and G. Pirsic, Duality for digital nets and its applications. *Acta Arith.*, 97, 173–182, 2001.
17. D. Potts, G. Steidl, and M. Tasche, Fast Fourier transforms for nonequispaced data: A tutorial. In “*Modern Sampling Theory: Mathematics and Applications*” ed. J.J. Benedetto and P. Ferreira, 249–274, Birkhäuser Boston, 2001.
18. C.M. Rader, Discrete Fourier transforms when the number of data samples is prime. *Proc. IEEE*, 56, 1107–1108, 1968.
19. I.H. Sloan and S. Joe, *Lattice methods for multiple integration*. Oxford University Press, Oxford, 1994.
20. I.M. Sobol’, On the distribution of points in a cube and the approximate evaluation of integrals. *U.S.S.R. Comput. Maths. Math. Phys.*, 7, 86–112, 1967.
21. I.M. Sobol’, *Multidimensional quadrature formulas and Haar functions*. Nauka, Moscow, 1969. (In Russian.)
22. I.M. Sobol’, Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Math. Comput. Simulation*, 55, 271–280, 2001.
23. G. Wahba, *Spline models for observational data*. SIAM, Philadelphia, 1990.
24. J.L. Walsh, A closed set of normal orthogonal functions. *Amer. J. Math.*, 55, 5–24, 1923.
25. X. Wang and K.T. Fang, The effective dimension and quasi-Monte Carlo integration. *J. Complexity*, 19, 101–124, 2003.
26. X. Wang and F.J. Hickernell, Randomized Halton sequences. *Math. Comput. Modelling*, 32, 887–899, 2000.
27. X. Wang and I.H. Sloan, Why are high-dimensional finance problems often of low effective dimension? *SIAM J. Sci. Comput.*, 27, 159–183, 2005.
28. X. Wang and I.H. Sloan, Efficient weighted lattice rules with applications to finance. *SIAM J. Sci. Comput.*, 28, 728–750, 2006.
29. X. Zeng, K.T. Leung and F.J. Hickernell, Error Analysis of Splines for Periodic Problems Using Lattice Design. *MCQMC2004* (ed. Niederreiter and Talay), 501–514, 2006.
30. P. Zinterhof, Über die schnelle Lösung von hochdimensionalen Fredholm-Gleichungen vom Faltungstyp mit zahlentheoretischen Methoden (On the fast solution of higher-dimensional Fredholm equations of convolution type by means of number-theoretic methods), Österreich. Akad. Wiss. Math.-Natur. Kl. Sitzungsber. II 196, no. 4–7, 159–169 (1987).

DEPARTMENT OF MATHEMATICS, HONG KONG BAPTIST UNIVERSITY, KOWLOON TONG, HONG KONG, P. R. CHINA

E-mail address: `kiliu@math.hkbu.edu.hk`

SCHOOL OF MATHEMATICS AND STATISTICS, UNIVERSITY OF NEW SOUTH WALES, SYDNEY 2052, AUSTRALIA

E-mail address: `josi@maths.unsw.edu.au`

DEPARTMENT OF APPLIED MATHEMATICS, ILLINOIS INSTITUTE OF TECHNOLOGY, ROOM E1-208, 10 W. 32ND STREET, CHICAGO, IL 60616, USA

E-mail address: `hickernell@iit.edu`